



VAASAN AMMATTIKORKEAKOULU
UNIVERSITY OF APPLIED SCIENCES

Amir Dirbaz

IOT-OHJAUSJÄRJESTELMÄ

Case Technobothnia

Tekniikka
2018

VAASAN AMMATTIKORKEAKOULU
Kone- ja tuotantotekniikka

TIIVISTELMÄ

Tekijä	Amir Dirbaz
Opinnäytetyön nimi	IOT-ohjausjärjestelmä
Vuosi	2018
Kieli	suomi
Sivumäärä	39 + 1 liitettä
Ohjaaja	Marko Rantasalo

Opinnäytetyön aiheena oli rakentaa etäohjattava pistorasia, joka toimii Arduinolla ja voidaan ohjata web-käyttöliittymän välityksellä.

Ajatuksena on ohjata muun muassa joulukuusen valoja, joita opiskelijat pääsevät omilla älypuhelimillaan sammuttamaan ja sytyttämään. Etäohjattava pistorasia toimisi myös Technobothnian WLAN-lähiverkossa.

Työssä käydään läpi projektin eri vaiheet tuotteen suunnittelusta toteutukseen asti, sekä käsitellään myös mikrokontrollerin ja Web-sovelluksen ohjelmointia, joilla Arduino saatiin yhdistettyä internetiin ja ohjattua sitä kautta.

Tulevaisuutta ajatellen kokonaisuutta tulisi kehittää paremmaksi mikrokontrollerista lähtien. ESP8266 on edullinen, mutta huonolaatuinen mikrokontrollerialusta, jota voi ainoastaan käyttää testaus tarkoituksissa, eikä varsinaisessa työssä. Opinnäytetyö tehtiin Vaasan ammattikorkeakoululle.

VAASAN AMMATTIKORKEAKOULU
UNIVERSITY OF APPLIED SCIENCES
Kone- ja tuotantotekniikka

ABSTRACT

Author	Amir Dirbaz
Title	IOT Control System
Year	2018
Language	Finnish
Pages	39 + 1 Appendix
Name of Supervisor	Marko Rantasalo

The thesis was made for Vaasan ammattikorkeakoulu. The purpose of the thesis was to implement a remote-controlled socket that works with Arduino and can be controlled via a web browser.

The idea was the control the Christmas lights from phones, so it can be easily turned on and off but also it would work as a remote-controlled socket in Technobothnia's WLAN-local area network.

The project involves various stages from designing to production. In addition to Internet network readiness, the thesis involved the programming of a microcontroller and a browser-based program where the printed circuit board was connected to and controlled via Internet.

In the future, the system could be improved on the part of the microcontroller. The ESP8266 is a cheap and lower-quality microcontroller platform that can be only used for testing purposes and not for actual work.

Keywords	Remote-controlled, Arduino, Esp8266, Local area network
----------	---

SISÄLLYS

TIIVISTELMÄ

ABSTRACT

LYHENTEET JA TERMIT	8
1 JOHDANTO	9
1.1 Työn tausta	9
1.2 Työn tavoite	9
2 KEHITYSYMPÄRISTÖ	10
2.1 Arduino	10
2.2 EPS8266-D1	10
2.3 Arduino-ohjelmisto	12
2.3.1 Luonnokset eli Sketches	12
2.4 Muut Osat	13
2.4.1 4-kanavainen relelähtö	13
2.4.2 LCD-näyttö	14
2.4.3 Pistorasia	15
2.4.4 Liitinpaneeli sulakepesällä	16
3 LAITEEN KOKOONPANO	17
3.1 Kotelo	17
3.2 Kytkentä Arduinoon	18
3.2.1 Relelähtö	18
3.2.2 LCD-näyttö	19
3.2.3 Painikkeet	19
3.3 Muut sähkökytkennät	20
3.3.1 Micro-USB-laturi	20
3.4 Sähköturvallisuuslaki	21
4 OHJELMOITI	22
4.1 Alkuasetusten määrittäminen	22
4.2 Ohjelmointi	23
4.2.1 Kirjasto	23

4.2.2	Muuttujat	24
4.2.3	Void setup	24
4.2.4	Void loop.....	25
4.2.5	Void HTTP.....	26
4.2.6	Void painike	26
5	KÄYTTÖLIITTYMÄ	27
5.1	Yhdistäminen langattomaan verkkoon	27
5.2	Ohjausyksikkö-välilehti	30
5.2.1	Pääsivu eli ohjausyksikkö-välilehti.....	30
5.2.2	Pistorasiat-välilehti.....	31
6	JOHTOPÄÄTÖKSET	32
	LÄHTEET	33

LIITTEET

KUVIO- JA TAULUKKOLUETTELO

Kuvio 1. ESP8266-D1.	10
Kuvio 2. Arduino-ohjelmisto IDE.	12
Kuvio 3. 4-kanavainen relelähtö.	13
Kuvio 4. LCD-näyttö.	14
Kuvio 5. Pistorasia pr1s.	15
Kuvio 6. Liitinpaneeli sulakepesällä.	16
Kuvio 7. 3D-malli laitteen kokoonpanosta.	17
Kuvio 8. Laitteen lopullinen kokoonpano.	17
Kuvio 9. KytKentäkaavio.	18
Kuvio 10. Muut sähkökytkennät.	20
Kuvio 11. Micro-USB-laturi ja pistorasia.	20
Kuvio 12. Preferences -ikkuna.	22
Kuvio 13. Board Manager -ikkuna.	23
Kuvio 14. WIFIManager-alkuasetukset.	27
Kuvio 15. WIFI-asetus-välilehti.	27
Kuvio 16. WIFIManager-asetus-välilehti.	28
Kuvio 17. WIFIManager -configure WIFI.	28
Kuvio 18. WIFIManager -yhdistyminen.	29
Kuvio 19. IP-osoite näytössä.	29
Kuvio 20. Ohjausyksikkö-välilehti.	30
Kuvio 21. Pistorasiat-välilehti.	31
Taulukko 1. ESP8266-D1 tekniset tiedot.	11
Taulukko 2. I/O porttien määrittäminen IDE-ohjelmassa.	11
Taulukko 3. Relelähdön tekniset tiedot.	13
Taulukko 4. LCD-näytön tekniset tiedot.	14
Taulukko 5. Pistorasian tekniset tiedot.	15
Taulukko 6. Liitinpaneelin tekniset tiedot.	16
Taulukko 7. Relelähtö Arduinon liitinnastaan.	18
Taulukko 8. Lcd-näytön kytkentä Arduinon liitinnastaan.	19
Taulukko 9. Painikkeet Arduinon liitinnastaan.	19

LIITELUETTELO

LIITE 1. Ohjelmakoodi

LYHENTEET JA TERMIT

Wi-Fi	Kaupallinen termi langattomalle lähiverkolle
WLAN	Local Area Network, langaton lähiverkko
Arduino	Arduino on avoimeen laitteistoon ja lähdekoodiin perustuva ohjelmoitava elektroniikka-alusta.
IDE	(Integrated Development Environment) Integroitu ohjelmointiympäristö
C/C++	Ohjelmointikieli
Html	Hypertext Markup Language, hypertekstin merkinäkieli.
IoT	Internet of Things (asioiden internet)
Mikrokontrolleri L	Laite jossa on mikroprosessori ja joitain muisti- ja liityntä-lohkoja, eli pieni tietokone
GPIO	General Purpose Input/Output; yleiskäyttöinen portti elektroniikassa
I2C	I2C-väylä, mikropiirien välisen tiedonsiirron väylä
I/O - liittimet I	I/O-liittimet
IP-protokolla	Internet-kerroksen protokolla.

1 JOHDANTO

1.1 Työn tausta

Opinnäytetyön aiheena oli rakentaa etäohjattava pistorasia, joka toimii Arduinolla ja voidaan ohjata web-käyttöliittymän välityksellä.

Alkuperäinen idea oli rakentaa itselle etäohjattava lämmitystolppa, jota pystyisin kotona säätämään ja käynnistämään etänä. Siihen olisi lisätty lämpöanturi ja kello, joiden avulla voitaisiin ajastaa ja säätää pitorasian päälläoloaikaa ilman lämpötilan mukaan.

C-kielen ohjelmointiosaamista ei ennestään ollut, mutta kiinnostus Arduinon ja IOT:a kohtaan olivat syitä miksi lähdettiin tähän työhön. Tämän työn aikana on tullut opittua todella paljon ohjelmoinnista ja IOT:sta. Useasti oppiminen riippuu siitä, kuinka hyvin ymmärtää ohjelmoinnin peruslogiikan. Ohjelmoitaessa on hallittava ohjelmointikoodia sekä omaksuttava sen rajoittamattomat mahdollisuudet. Yleisesti ottaen ohjelmistokielet ovat hyvin pitkälti samantyyllisiä, mutta kirjoitusasut vaihtelevat.

1.2 Työn tavoite

Työn tavoitteena on ensisijaisesti ohjata joulukuusen valoja, joita opiskelijat pääsevät omilla puhelimillaan sammuttamaan ja käynnistämään etänä. Työn ideana on rakentaa etäohjattava pistorasia, joka toimisi Technobothnian WLAN-lähiverkossa.

Tässä opinnäyteyössä käydään yksityiskohtainesti läpi, miten päädyttiin suunnittelemaan sekä toteuttamaan etäohjattava pistorasia.

2 KEHITYSYMPÄRISTÖ

2.1 Arduino

Arduino on avoimeen laitteistoon ja ohjelmistoon perustuva mikrokontrollerialusta sekä ohjelmistoympäristö. Arduinon toimintaperiaate on ohjata sisään- ja ulostuloja (INPUT & OUTPUT), joihin voidaan kytkeä sensoreita ja toimilaitteita. /7/

2.2 ESP8266-D1

ESP8266-D1 on langaton 802.11 (WIFI) mikrokontrollerialusta, joka on yhteensopiva Arduino IDE:n kanssa (Kuvio 1.). Siihen on asennettu valmiiksi ESP8266-WIFI-moduuli ja se on kilpailevaa alustaa halvempi. ESP8266-D1 perustuu hyvin pitkälti Arduino Leonardo mikrokontrolleriin ja näin olleen voidaan hyödyntää Arduinon soveltuvia varaosia ja moduuleja. /2/



Kuvio 1. ESP8266-D1.

Tekniset tiedot:

Taulukko 1. ESP8266-D1 tekniset tiedot.

Mikrokontrolleri:	ESP-8266
Kellotaajuus:	80MHz/160MHz
Käyttöjännite:	3.3V
Digital I/O Pins:	11 kpl
Analog Input Pins:	1 kpl
Syöttöjännite:	5V - 24VDC
Mitat:	68.6mm x 53.4mm
Paino:	25g

I/O Liitinnastas:

Taulukko 2. I/O porttien määrittäminen IDE-ohjelmassa.

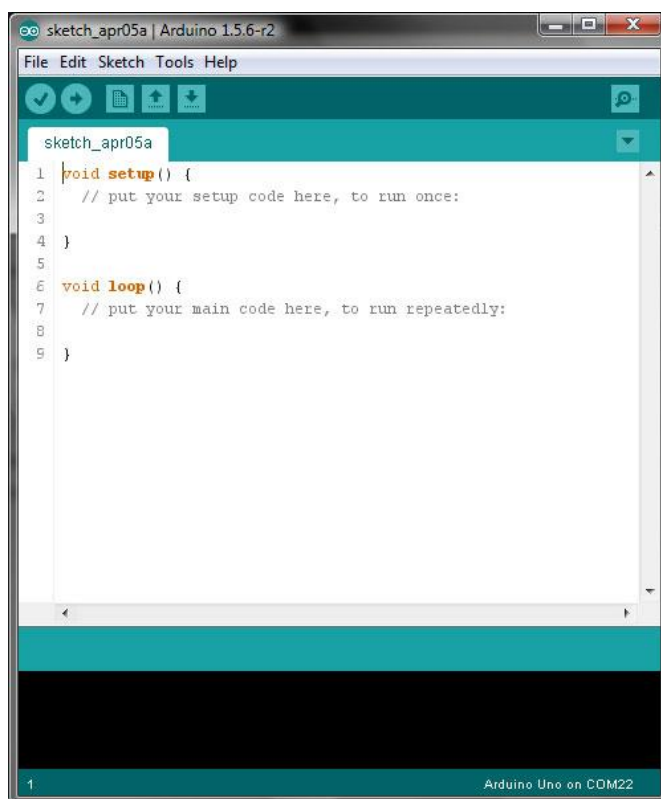
Board Pin	Function	ESP8266 Pin
TX	TXD	TXD
RX	RXD	RXD
A0	Analog in	A0
D0	I/O	GPIO16
D1	I/O, SCL	GPIO5
D2	I/O, SDA	GPIO4
D3	I/O, 10k pull-up	GPIO0
D4	I/O, 10k pull-up	GPIO2
D5	I/O, SCK	GPIO14
D6	I/O, MISO	GPIO12
D7	I/O, MOSI	GPIO13
D8	I/O, 10k pull-up	GPIO15
GND	Ground	GND
5V	5V	
3V3	3.3V	3.3V
RST	Reset	RST

2.3 Arduino-ohjelmisto

Arduinon ohjelmaympäristö perustuu ohjelmointikieleen (integrated development environment) ”IDE ” (Kuvio 2.). Ohjelmiston avulla voidaan ohjelmoida ja määrittää laitteen toimintoja. Arduino-ohjelmistossa ohjelmoidaan C-kielellä. /1/

2.3.1 Luonnokset eli Sketches

Arduino-ohjelmiston (IDE) avulla kirjoitettuja ohjelmia kutsutaan luonnoksiksi. Nämä luonnokset on kirjoitettu tekstieditorissa ja ne tallennetaan tiedostopäätteeseen. Arduino-ohjelma näyttää myös virheitä ja kehotuksia koodin korjaamiseen. Työkalupalkin painikkeilla voi tarkistaa, ladata ohjelmia, luoda, avata ja tallentaa luonnoksia. /1/

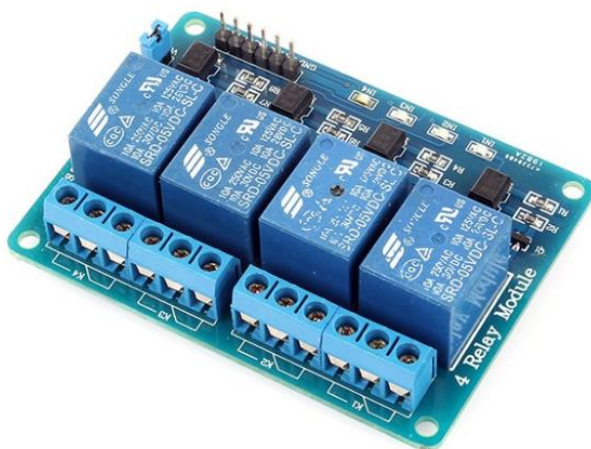


Kuvio 2. Arduino-ohjelmisto IDE.

2.4 Muut Osat

2.4.1 4-kanavainen relelähtö

Relelähdoillä voidaan ohjata korkeaännitteisiä laitteita matalalla jännitteellä. Relelähtö on sähköisesti ohjattava sähkömekaaninen kytkin. Sähkömagneetti aktivoidaan pienellä jännitteellä, esimerkiksi 5 V, joka vetää releen liikkuvan kärjen kiinni paikallaan olevaan kontaktiin, tai irrottaa sen siitä, riippuen releen rakenteesta (Kuvio 3.). /6/



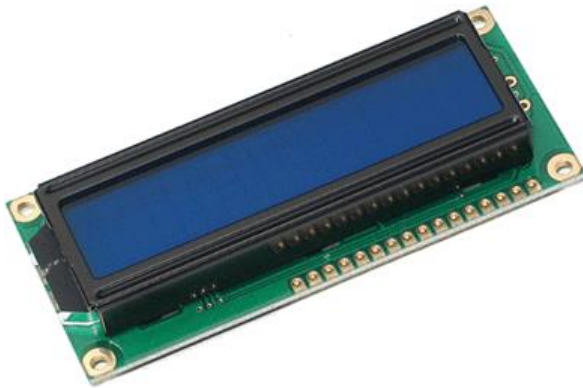
Kuvio 3. 4-kanavainen relelähtö.

Tekniset tiedot:

Taulukko 3. Relelähdon tekniset tiedot.

Malli:	Songle SRD-05VDC-SL-C
Kuormitettavuus:	10 A (AC 250 V / DC 30 V)
Ohjausjännite:	5 V (toimii myös 3.3 voltilla)
Tuloa:	IN1, IN2, IN3, IN4, GND, VCC
Lähtöä:	NO, NC, COMMON
Mitat:	43x17 mm

2.4.2 LCD-näyttö



Kuvio 4. LCD-näyttö.

Tekniset tiedot:

Taulukko 4. LCD-näytön tekniset tiedot.

Malli:	HJ1602A & PCF8574T
Käyttöjännite:	2.5 - 6 V
Merkkejä:	2 riviä, 16 merkkiä rivillä (yhteensä 32)
Tulopinnit:	GND, VCC, SDA, SCL
Näyttöalueen koko:	64.5 x 16 mm
Mitat:	80 x 36 x 12 mm

2.4.3 Pistorasia



Kuvio 5. Pistorasia pr1s.

Taulukko 5. Pistorasian tekniset tiedot.

Malli:	PR1S
Jännitekesto:	220-250 Vac 50-60 Hz
Virtakesto:	16 A
Tiiveysluokka:	IP 54
Runkomateriaali:	nylon
Kosketinmateriaali:	messinki
Väri:	sininen
Liitäntä:	ruuviliitäntä
Johdinkoko:	1,5-4,0 mm ²
Napaluku:	2 + suojamaa
Polariteetti:	naaras
Asennusaukko:	43 mm
Paneelin mitat (LxK):	50,5x50,5 mm

2.4.4 Liitinpaneeli sulakepesällä



Kuvio 6. Liitinpaneeli sulakepesällä.

Taulukko 6. Liitinpaneelin tekniset tiedot.

Malli:	BZH01-01
Standardi:	IEC 320 / EN 60320-1
Jännitekesto:	250 Vac
Virtakesto:	A 10
Liittimen napaluku:	3 (2 + 1)
Polariteetti:	uros
Sulakekoko:	5x20 mm 10 A maks.
Kiinnitys:	SNAP-IN-pikakiinnitys
Asennusaukko (LxK):	60,3x34,4 mm
Paneelin paksuus:	1-3 mm
Liitäntä:	6,3 mm pikaliittimet
Kytkin:	1 x ON-OFF musta

3 LAITEEN KOKOONPANO

3.1 Kotelo

Tämän työn suunnittelun tavoitteena on ohjata neljää pistorasia langattomasti. Projektiin tarvittiin kotelo, johon pistorasiat voitiin asentaa. Sopivan muovikotelo löytyi Vaasan elektroniikkakeskuksen sivustolta. Ideana oli käyttää mahdollisimman pientä koteloä. Suunnittelussa käytettiin Siemens NX-ohjelmistoa ja muutaman kotelomallinnuksen jälkeen sen avulla löytyi sopiva kotelo (Kuvio 7.).



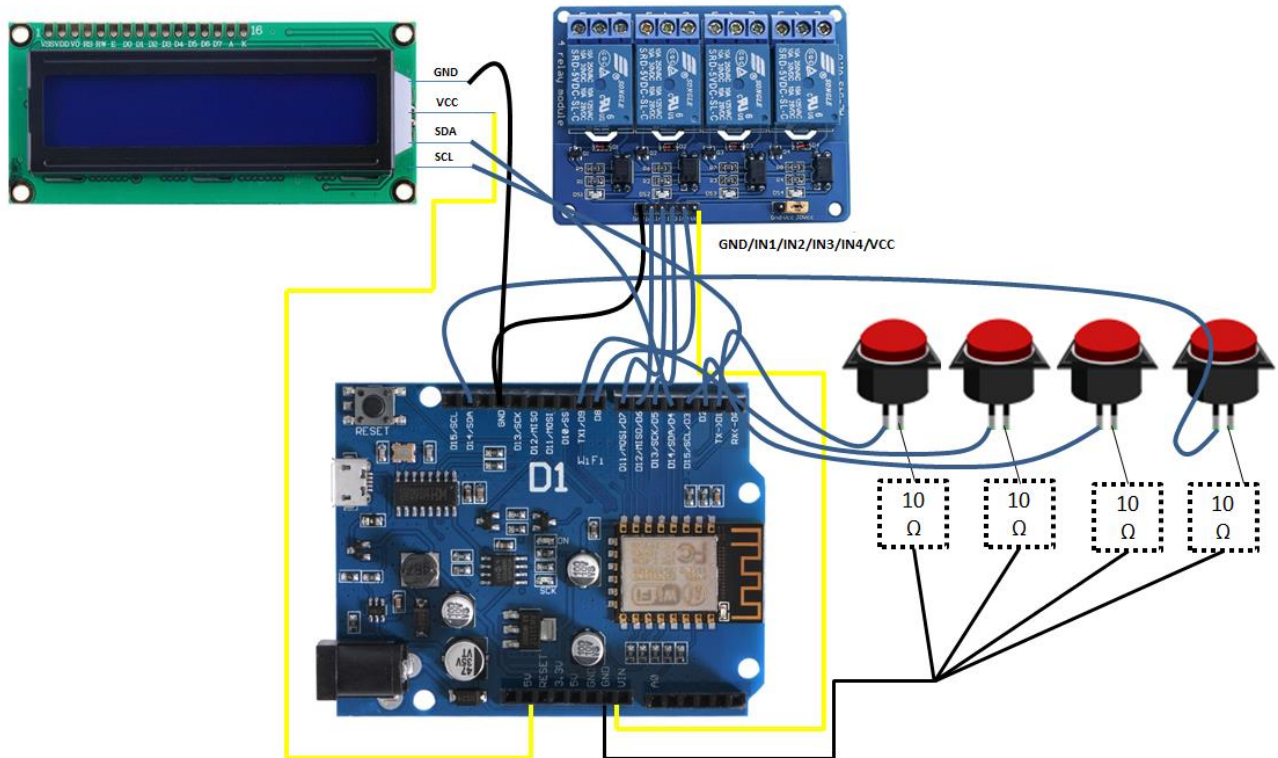
Kuvio 7. 3D-malli laitteen kokoonpanosta.



Kuvio 8. Laitteen lopullinen kokoonpano.

3.2 Kytkeä Arduinoon

Arduinon kytkentä on haastava, koska sisääntulojen merkinnät piirilevyssä ja ohjelmassa olivat erilaiset (Kuvio 9.).



Kuvio 9. Kytkeäkaavio.

3.2.1 Relelähdtö

Relelähdtöön kytkeä sisääntuloon:

Taulukko 7. Relelähdtö Arduinon liittinnastaan.

GND	Maadoitus
IN1	GPIO12
IN2	GPIO13
IN3	GPIO14
IN4	GPIO02
VCC	Jännite 5V

3.2.2 LCD-näyttö

LCD-näytön kytkentä sisääntuloon:

Taulukko 8. Lcd-näytön kytkentä Arduinon liitinnastaan.

GND	Maadoitus
VCC	Jännite 5V
SDA	GPIO04
SCL	GPIO05

3.2.3 Painikkeet

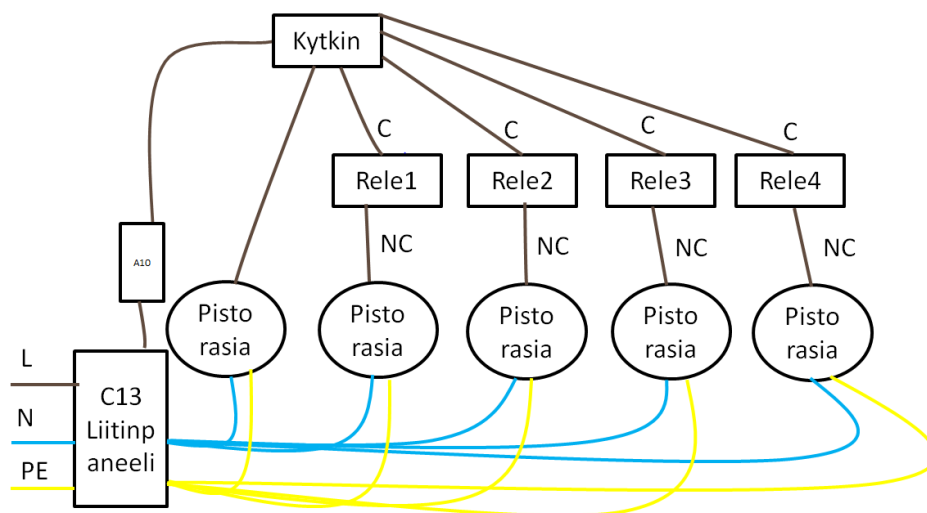
Arduinossa painikkeiden kanssa pitää kytkeä myös vastus, koska Arduinossa tulon jännitearvo voi vaihdella nollan ja viiden voltin välillä. Tässä työssä käytettiin 10 Ω 5 % vastusta, eli vastuksen tarkkuus on 9.5 – 10.5 Ω välillä.

Taulukko 9. Painikkeet Arduinon liitinnastaan.

Painike 1	GPIO01
Painike 2	GPIO00
Painike 3	GPIO03
Painike 4	GPIO04

3.3 Muut sähkökytkennät

Sähköjohtojen kytkennässä otettiin huomioon tunnusvärit ja kytkettiin ne sen mukaan. Boksiin kytkettiin tavallinen C13-virtajohto (Kuvio 10.).



Kuvio 10. Muut sähkökytkennät.

3.3.1 Micro-USB-laturi

Kotelon sisälle kytkettiin pistorasia, joka on tarkoitettu mikrokontrollerin laturille (Kuvio 11.). Arduino käyttää 5 V:n jännitettä.



Kuvio 11. Micro-USB-laturi ja pistorasia.

3.4 Sähköturvallisuuslaki

Sähkölaitteiden korjaus ja huoltotöitä, sekä sähkölaitteistojen rakennus- ja korjaustöitä saa tehdä, kunhan täyttää laissa mainitun edellytyksen. Lain mukaan sähkölaitteistojen käytöstä pitää tehdä valvontaa varten ilmoitus sähköturvallisuusviranomaisille.

Sähkölaitteistojen rakennus-, korjaus-, huolto- ja käyttöönoton saa tehdä:

- Henkilö, jolla on riittävä kelpoisuus tai muuten riittävä ammattitaito
- Töiden tekemiseen kannalta tarpeelliset tilat, työvälineet sekä sähköturvallisuutta koskevat säännökset ja määräykset. /8/

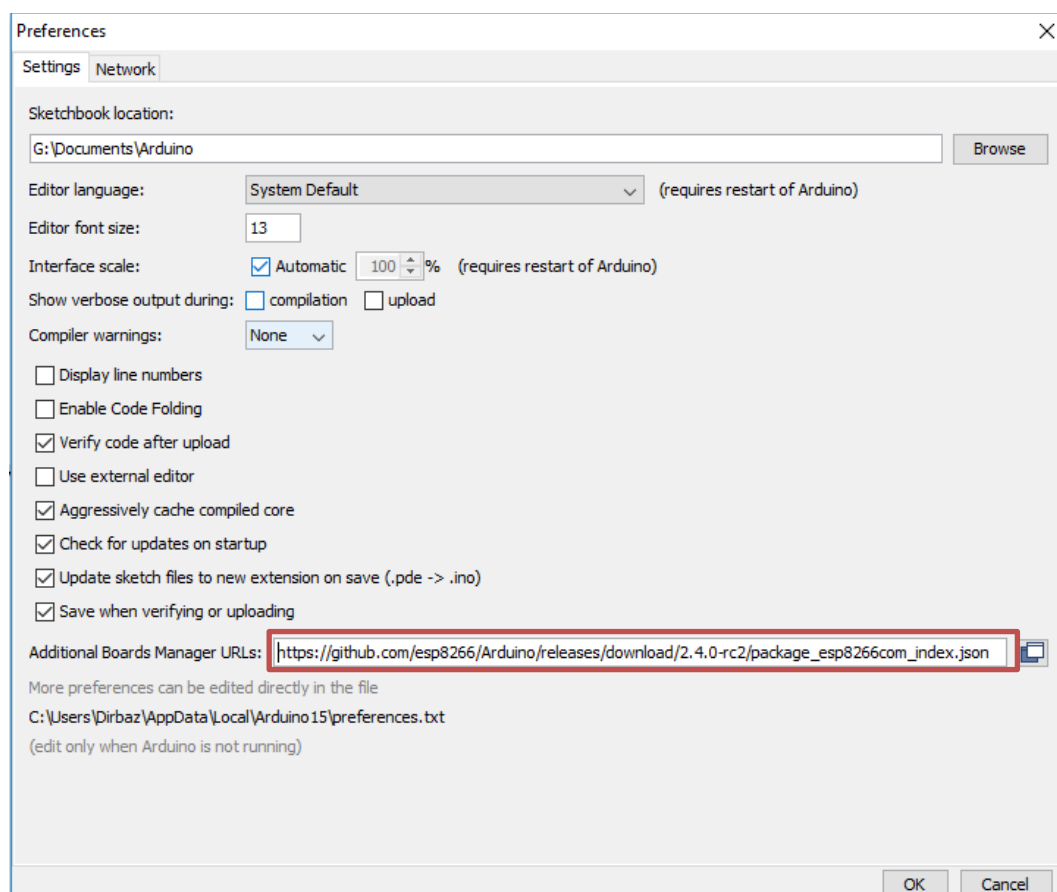
4 OHJELMOITI

4.1 Alkuasetusten määrittäminen

Arduino IDE-ohjelmassa määritetään jokaiselle alustalle oma asetus. Työssä käytettyä mikrokontrollerialustaa ei löytynyt valmiiksi ohjelmasta, jonka vuoksi se jouduttiin lataamaan verkosta. /4/

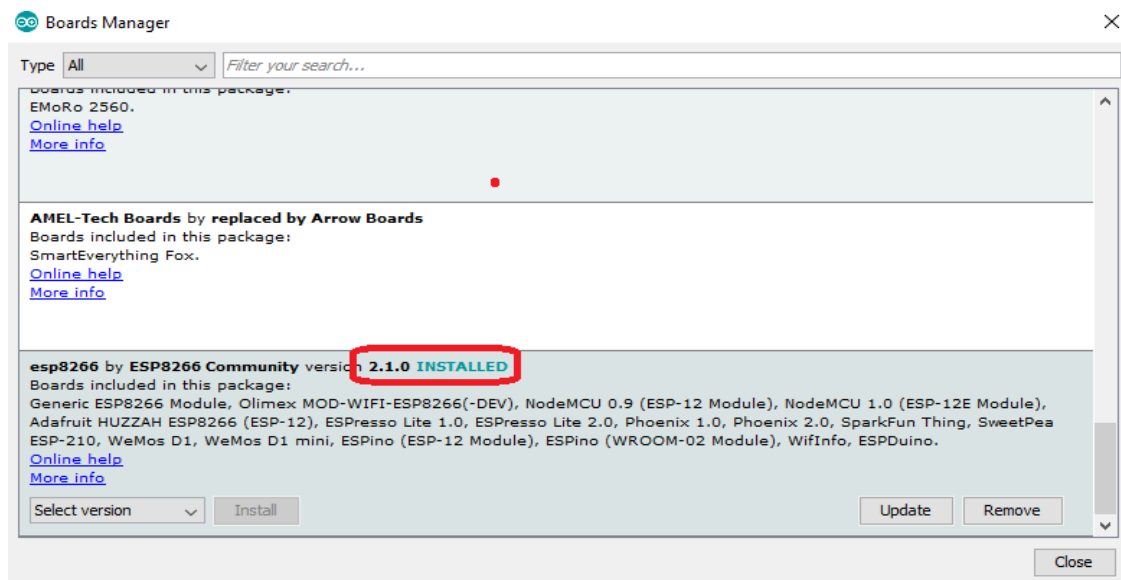
Ensin avataan Arduino IDE-ohjelma, jonka jälkeen mennään File > Preferences-ikkunaan ja sen jälkeen lisätään linkki, kuten kuvasta selviää (Kuvio 12.).

Linkin avulla ohjelma pystyy hakemaan oikean mikrokontrollerialustan verkosta.
”https://github.com/esp8266/Arduino/releases/download/2.4.0-rc2/package_esp8266com_index.json”



Kuvio 12. Preferences -ikkuna.

Tämän jälkeen mennään Tools > Board > Board Manager- ikkunaan ja haetaan, sekä asennetaan esp8266 (Kuvio 13.).



Kuvio 13. Board Manager -ikkuna.

4.2 Ohjelmointi

4.2.1 Kirjasto

Kirjastot ovat kokoelma koodeja, joiden tehtävänä on helpottaa yhteyden muodostamista anturin, näytön ja moduulin välillä. Tässä työssä käytetään kuutta eri kirjastoa:

- DoubleResetDetector – kirjaston avulla mikrokontrollerialustassa olevaa painiketta voidaan hyödyntää omiin tarkoituksiin. Kun painetaan kolme kertaa peräkkäin, WIFI-asetuksen parametrit ohjelmassa nollautuvat. /5/
- ESP8266WIFI – kirjaston avulla voidaan hyödyntää mikrokontrollerialustassa olevaa WIFI-moduulia.
- ESP8266WebServer – kirjaston avulla voidaan muuttaa mikrokontrollerialustapalvelinta, johon toiset laitteet voivat ottaa yhteyttä.
- DNSServer – Yksinkertainen DNS-palvelin, jonka avulla voidaan toteuttaa STA (A client device) ja AP-palvelua (AccessPoint).

- WIFIManager – kirjaston avulla voidaan muuttaa WIFI-asetusten parametreja ilman, että tarvitsee kytkeä mikrokontrollerialusta tietokoneeseen. /6/
- LiquidCrystal_I2C – kirjaston avulla voidaan ohjata lcd-näyttömoduulia mikrokontrollerialustassa.

4.2.2 Muuttujat

Muuttujat ovat nimettyjä arvoja, joita voidaan käyttää ohjelmassa. Niihin voidaan tallentaa tietoja ja niitä voidaan myöhemmin käyttää ohjelmassa niin usein kun tarvitaan.

- rele1 – muuttuja on tyypiltään kokonaisluku, joka saa arvoksi luvun 12. Mikrokontrollerialustassa kytkettiin Relekortti IN1 liitinnastaan 12.
- painike1 – muuttuja on tyypiltään kokonaisluku, joka saa arvoksi luvun 1. Mikrokontrollerialustassa kytkettiin ensimmäinen painike liitinnastaan 1.
- Server (80) – muuttujalle määritellään portti numero 80, joka on oletusarvo web-palvelimen porttinumeroksi.
- LiquidCrystal_I2C lcd (0x27, 16, 2) - muuttujalle annettiin arvot
 - 0x27 – tietokoneiden merkistö
 - 16 – merkkien määrä
 - 2 – rivien määrä
- DoubleResetDetector res – muuttujaa kutsutaan ohjelmassa silloin kun halutaan nollata WIFI-asetusten parametreja.

4.2.3 Void setup

Setup () -funktio kutsutaan luonnokseksi, eli sketch:ksi, käynnistyksessä. Sen avulla voidaan mm. alustaa muuttujat ja liitinnastat, sekä aloittaa kirjastot. Setup () -funktio suoritetaan vain kerran. Siinä tehdään kertaluontoiset alustukset sekä kaikki sellaiset koodit, jotka tulee suorittaa ohjelman alussa. /6/

Rivillä 21 avataan sarjaportti nopeudella 115,200 kbit/s, joka kirjoittaa tekstiä luettavaksi sarjakonsolilta.

Rivillä 22–28 aloitetaan lcd -muuttuja, joka saa lcd-näytön päälle. Sen jälkeen se kirjoittaa” Amir Dirbaz ”-tekstin näytön oikeaan yläkulmaan. Ohjelma odottaa viisi sekuntia ja sen jälkeen se tyhjentää näyttöä.

Rivillä 29–35 kutsutaan kirjastosta WIFIManager-muuttuja, samalla ohjelma odottaa käyttäjän mahdollisesti painavan mikrokontrollerialustassa olevaa painiketta. Jos mitään muutosta ei ole havaittu, ohjelma näyttää näytössä "SMARTBOX 123456789BOX" -tekstiä, joka on oletus asetus WIFille. Sen jälkeen se odottaa, että käyttäjä ottaa yhteyden WLANin avulla laiteeseen. Yhteyden muodostumisen jälkeen näytölle tulee ”YHDISTETTY” -teksti.

Rivillä 41–53 laitteen yhdistymisen jälkeen ohjelmaa määrittää

- pinMode (rele1, OUTPUT), Annetaan rele1:lle arvo ”tulostustila” eli output.
- digitalWrite (rele1, HIGH), Laittaa liitinnastan 12 päälle -tilaan (HIGH eli +5 V).
- pinMode (painike1, INPUT_PULLUP), Annetaan painike1:lle arvo ”lukutilaan” eli input.

Rivillä 56–58 aloitetaan serverin käyttö, jonka muuttujan nimi on HTTP.

4.2.4 Void loop

Toinen olennainen funktio Setup () -funktion jälkeen on Loop (), jonka tehtävänä on alustaa ja määrittää ohjelman toiminnallisuusarvoja. Loop () -funktion koodi toistetaan niin kauan kun mikrokontrollerialustassa on virtaa, eli toistetaan annettua ohjelmakoodia, kunnes annettu ehto ei enää evaluoidu todeksi. /6/

Rivillä 63–66 haetaan painike-funktiota, näytetään lcd-näytössä IP-osoitetta, sekä testataan serverin yhteydet.

4.2.5 Void HTTP

Suurin osa ohjelman suoritusajasta kuuluu toistaen tätä funktiota. Kaikki laitteen toiminta perustuu tämän funktion sisälle, joten se on todella merkittävää osa ohjelman toimintaa.

Rivillä 70-76 luodaan muuttuja ”pistoke 1”, joka on tietotyypiltään merkkijono. Sen avulla voidaan kysellä parametreja HTTP-pyyntöön. Tätä varten pyydetään args-menetelmä Server-objektiin. Tämä menetelmä palauttaa kyselyn parametrin, joka välitetään HTTP-pyyntöön ja käynnistää käsittelytoiminnon suorittamisen. Aluksi annetaan ehdoksi, että ”pistoke1” on ”off”, joten mikrokontrollerialusta kirjoittaa liitinnastan ”rele1” arvoksi HIGH, joka tarkoittaa sitä, että ”rele1” on pois päältä. Päällekytkeminen tapahtuu samalla tavalla, mutta liitinnastan ”rele1” arvoksi tulee LOW.

Rivillä 98 luodaan muuttuja (Web), joka on tietotyypiltään merkkijono, joka kääntää tekstin html-kielille.

Rivillä 100–260 suunnitellaan Web-sovellusta, jonka ohjelma näyttää selaimessa. Rivillä 119–133 luodaan ympyräkuvio, joka näyttää selaimessa vihreää, jos pistoke on päällä, tai punaista jos se on pois päältä. Rivillä 195–205 luodaan painike ”OFF”, joka sammuttaa pistokkeen lähettämällä pistoke 1=off -pyynnön linkin avulla. Päälle kytkeminen tapahtuu samalla tavalla. Ensin luodaan painike nimeltään ”ON” ja annetaan sille pistoke 1 = ON pyyntö linkin avulla. Ohjelmassa on myös IF-lause -ehto, jos toinen painikkeesta on aktiivinen, niin se menee piiloon. Lopussa, rivillä 261, send-metodin avulla lähetetään Server-objekti.

4.2.6 Void painike

Painike-funktiolla tarkistetaan painikkeen olotila. Painike on aktiivisena, kun se kirjoittaa mikrokontrollerialustan liitinnastan ”rele1” arvoksi LOW.

5 KÄYTTÖLIITTYMÄ

5.1 Yhdistäminen langattomaan verkkoon

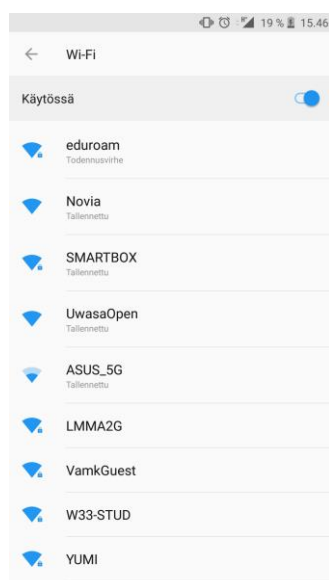
Laitteen yhdistämiseksi langattomaan verkkoon tarvitaan langattoman verkon nimi ja salasana. Nämä tiedot löytyvät laitteen lcd-näytöstä, kun ensimmäistä kertaa laitetta käynnistetään tai WIFI-asetukset nollataan.

1. Laite kytketään päälle ja odotetaan, että laitteen näyttöön ilmestyy WLAN-verkon oletusnimi ja -salasana (Kuvio 14.).



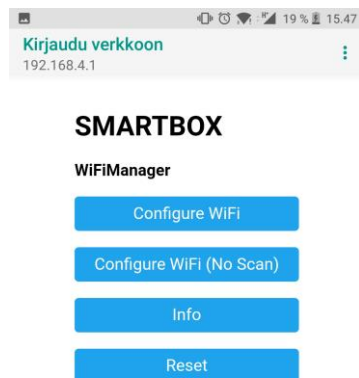
Kuvio 14. WIFIManager-alkuasetukset.

2. Avataan omasta laitteesta (esim. tietokone, tabletti tai älypuhelin), WIFI-asetukset -välilehti. Liitytään SMARTBOX verkkoon (Kuvio 15.).



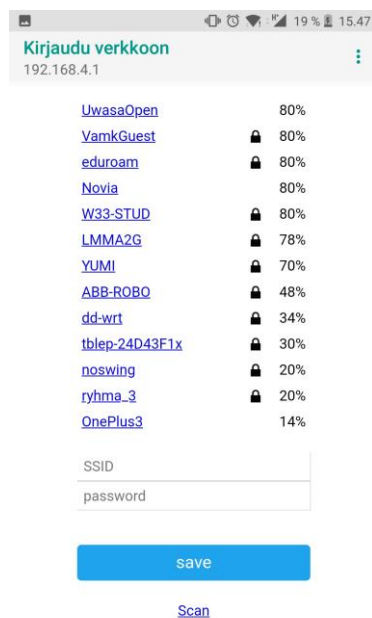
Kuvio 15. WIFI-asetus-välilehti.

3. Avataan selaimesta osoite 192.168.4.1 ja valitaan Configure WIFI (Kuvio 16.).



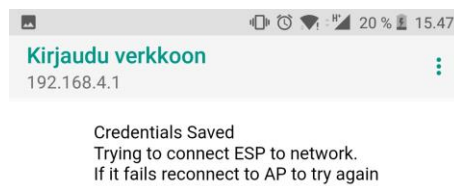
Kuvio 16. WiFiManager-asetus-välilehti.

4. Tässä kohdassa valitaan verkko, johon halutaan laitteen yhdistyvään (Kuvio 17.).



Kuvio 17. WiFiManager -configure WIFI.

5. Nyt WIFI-asetukset on tallennettu mikrokontrollerialustalle (Kuvio 18.).



Kuvio 18. WIFIManager -yhdistyminen.

6. Kun yhteys onnistuu, niin laitteen lcd-näyttöön ilmestyy IP-osoite (Kuvio 19.).



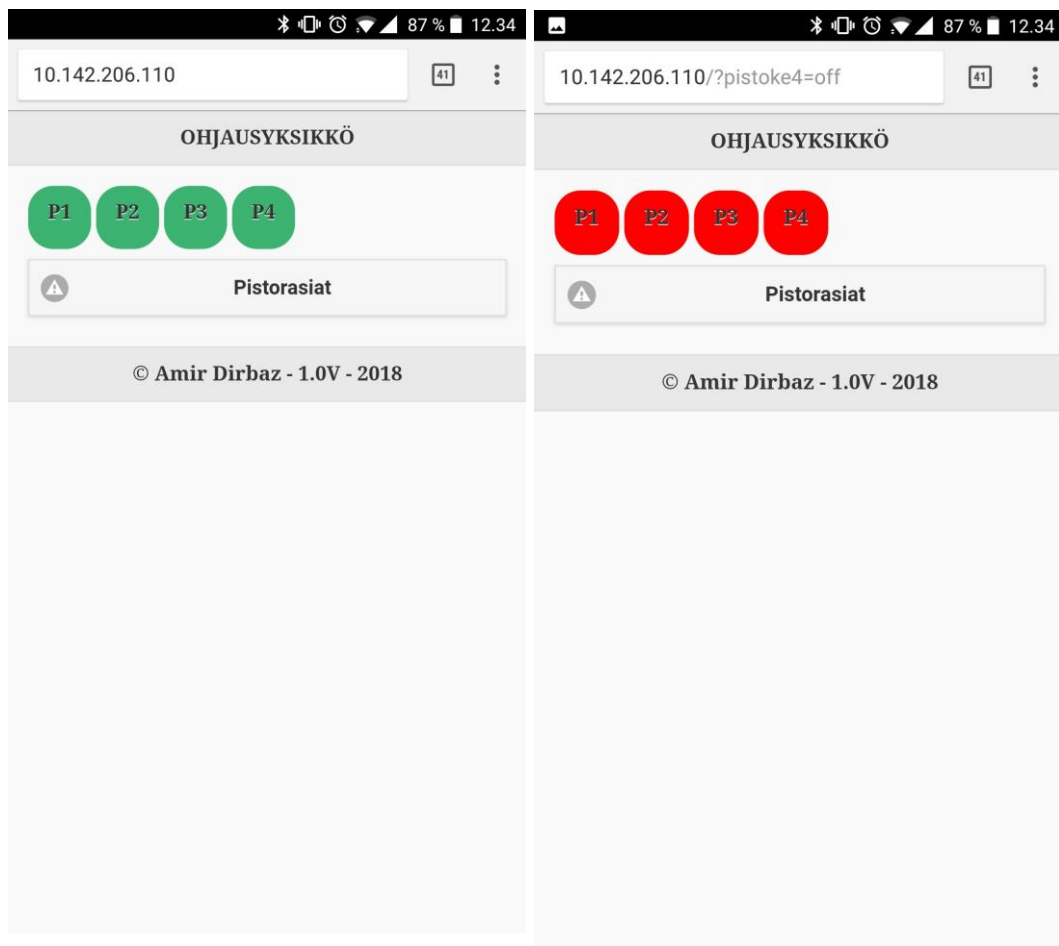
Kuvio 19. IP-osoite näytössä.

5.2 Ohjausyksikkö-välilehti

Laitteen yhdistymisen jälkeen avataan selain ja syötetään IP-osoite.

5.2.1 Pääsivu eli ohjausyksikkö-välilehti

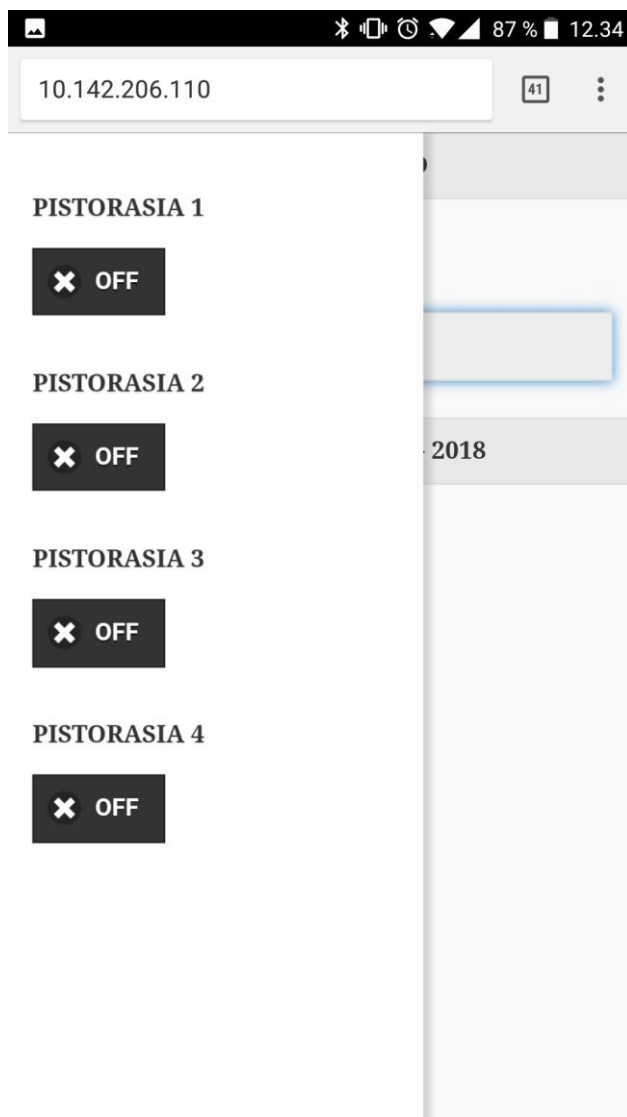
Pääsivulla nähdään aktiiviset pistorasiat, sekä linkki pistorasioiden ohjaamiseen (Kuvio 20.).



Kuvio 20. Ohjausyksikkö-välilehti.

5.2.2 Pistorasiat-välilehti

Pistorasioiden välilehdellä voidaan ohjata pistorasioita painikkeiden avulla. Kun pistorasia on päällä, ”OFF” -painike korvaa ”ON” -painikkeen (Kuvio 21.).



Kuvio 21. Pistorasiat-välilehti.

6 JOHTOPÄÄTÖKSET

Tämän opinnäytetyön toteuttaminen auttoi ymmärtämään laajemmin tietoteknillisiä kokonaisuuksia, sekä erilaisia ohjelmointikieliä. Työn aikana tuli opittua paljon uusia asioita Arduinon ohjaamisesta, sekä parempaa ymmärrystä toiminnallisten kokonaisuuksien hallinnasta. Opinnäytetyön tuloksena saatiin toimiva etäohjattava pistorasiaryhmä, jota voidaan hyödyntää monipuolisesti sekä kotona että teollisuudessa. Omasta näkökulmastani työ oli todella mielenkiintoinen ja haasteellinen.

Tulevaisuutta ajatellen kokonaisuutta tulisi kehittää paremmaksi mikrokontrollerista lähtien. ESP8266 on edullinen, mutta huonolaatuinen mikrokontrollerialusta, jota voi käyttää ainoastaan testaustarkoituksissa, eikä varsinaisessa työssä. Parempi vaihtoehto olisi Arduino Yun, jonka avulla varmaankin pääsisi parhaaseen lopputulokseen, koska se on luotettavampi ja monipuolisempi kehitysalusta. Näin päästäisiin parantamaan laitteen toimivuutta huomattavasti. Jatkokehityksenä voisi lisätä kellon sekä ajastus-ratkaisun.

Projektin toteuttaminen kesti noin kuukauden, ja kirjoitusosuus noin kolme kuukautta. Tutkimustyö ja Arduinon toiminnallisuuden selvittämiseen meni yli kuusi kuukautta. Aikataulussa silti pysyttiin, sillä projektin tulisi olla valmiina ennen joululomaa. Opiskelijat eivät kuitenkaan pääse testaamaan laitetta kiinteän IP-osoitteen puutteen takia.

LÄHTEET

/1/ Arduino. 2018. Arduinon virallinen kotisivu. Viitattu 23.1.2018 <http://www.arduino.cc/>

/2/ Ihmevekotin. 2018. Kehitysalusta verkkokauppa. Viitattu 25.1.2018 <http://www.ihmevekotin.fi/>

/3/ StarElec Oy. 2018. Elektroniikka-alan yritys. Viitattu 27.1.2018 <https://www.vekoy.com/>

/4/ ESP8266. 2018. ESP8266 liittäminen Arduino ympäristöön. Viitattu 28.1.2018 <https://github.com/esp8266/Arduino>

/5/ DoubleResetDetector. 2018. Kaksois-painike kirjasto. Viitattu 28.1.2018 <https://github.com/datacute/DoubleResetDetector>

/6/ WIFIManager. 2018. WIFIManager kirjasto. Viitattu 28.1.2018 <https://github.com/tzapu/WIFIManager>

/6/ Karvinen, T & K. 2010. Sulautetut. Helsinki. Readme.fi.

/7/ Banzi, M. 2011. Arduino perusteista hallintaan. O'Reilly.

/8/ Tiainen, E. 2014. Sähköasennusopas. Helsinki. Sähkö- ja teleurakoitsijaliitto STUL ry.

LIITE 1 ohjelmakoodi

```

1. #include <DoubleResetDetector.h>
2. #include <ESP8266WiFi.h>
3. #include <ESP8266WebServer.h>
4. #include <DNSServer.h>
5. #include <WIFIManager.h>
6. #include <LiquidCrystal_I2C.h>
7. #define DRD_TIMEOUT 10
8. #define DRD_ADDRESS 0
9. int rele1 = 12;
10. int rele2 = 13;
11. int rele3 = 14;
12. int rele4 = 02;
13. int painike1 = 1;
14. int painike2 = 0;
15. int painike3 = 3;
16. int painike4 = 4;
17. ESP8266WebServer server(80);
18. LiquidCrystal_I2C lcd(0x27, 16, 2);
19. DoubleResetDetector res(DRD_TIMEOUT, DRD_ADDRESS);
20. void setup() {
21.   Serial.begin(115200);
22.   lcd.begin();
23.   lcd.setCursor(0,0);
24.   lcd.print("  Amir Dirbaz");
25.   lcd.setCursor(0,1);
26.   lcd.print("  VAMK 2018");
27.   delay(2000);
28.   lcd.clear();
29.   WIFIManager WIFIManager;
30.   if(res.detectDoubleReset()){
31.     WIFIManager.resetSettings();
32.   }
33.   delay(500);
34.   ip();
35.   WIFIManager.autoConnect("SMARTBOX", "123456789BOX");
36.   lcd.clear();
37.   lcd.print("YHDISTETTY...");
38.   delay(500);
39.   lcd.clear();
40.   Serial.println("YHDISTETTY");
41.   if (WiFi.waitForConnectResult() == WL_CONNECTED) {
42.     pinMode(rele1 , OUTPUT);
43.     digitalWrite(rele1 , HIGH);
44.     pinMode(rele2 , OUTPUT);
45.     digitalWrite(rele2 , HIGH);
46.     pinMode(rele3 , OUTPUT);
47.     digitalWrite(rele3 , HIGH);
48.     pinMode(rele4 , OUTPUT);
49.     digitalWrite(rele4 , HIGH);
50.     pinMode(painike1 , INPUT_PULLUP);
51.     pinMode(painike2 , INPUT_PULLUP);
52.     pinMode(painike3 , INPUT_PULLUP);
53.     pinMode(painike4 , INPUT_PULLUP);
54.     lcd.backlight();
55.     lcd.clear();
56.     server.on ( "/", HTTP );
57.     server.onNotFound ( HTTP );
58.     server.begin();
59.   }
60.   res.stop();
61. }
62. void loop() {
63.   painik1();
64.   lcd.setCursor(1,0);

```

```

65. lcd.print(WIFI.localIP());
66. server.handleClient();
67. delay(10);
68. }
69. void HTTP(void) {
70. String pistoke1 = server.arg("pistoke1");
71. if (pistoke1 == "off") {
72.     digitalWrite(rele1, HIGH);
73. }
74. else if (pistoke1 == "on") {
75.     digitalWrite(rele1, LOW);
76. }
77. String pistoke2=server.arg("pistoke2");
78. if (pistoke2 == "off"){
79.     digitalWrite(rele2, HIGH);
80. }
81. else if (pistoke2 == "on") {
82.     digitalWrite(rele2, LOW);
83. }
84. String pistoke3=server.arg("pistoke3");
85. if (pistoke3 == "off"){
86.     digitalWrite(rele3, HIGH);
87. }
88. else if (pistoke3 == "on") {
89.     digitalWrite(rele3, LOW);
90. }
91. String pistoke4=server.arg("pistoke4");
92. if (pistoke4 == "off"){
93.     digitalWrite(rele4, HIGH);
94. }
95. else if (pistoke4 == "on") {
96.     digitalWrite(rele4, LOW);
97. }
98. String web = "";
99. web =
100.     "<html lang=\"fi\">\
101.         <head>\
102.             <meta charset=\"UTF-8\">\
103.             <meta name=\"viewport\" content=\"width=device-width, initial-scale=1\">\
104.             <link rel=\"stylesheet\" href=\"https://code.jquery.com/mobile/1.4.5/jquery.mo-
105.             bile-1.4.5.min.css\">\
106.             <script src=\"https://code.jquery.com/jquery-1.11.3.min.js\"></script>\
107.             <script src=\"https://code.jquery.com/mobile/1.4.5/jquery.mobile-
108.             1.4.5.min.js\"></script>\
109.             <style>\
110.                 h1 {font-family: \"Times New Roman\", Georgia, Serif; font-size: 16px;}\
111.                 p {font-family: \"Times New Roman\", Georgia, Serif; font-size: 16px;}\
112.             </style>\
113.         </head>\
114.         <body>\
115.             <div data-role=\"page\" id=\"pageone\">\
116.                 <div data-role=\"header\">\
117.                     <h1>OHJAUSYKSIKKÖ</h1>\
118.                 </div>\
119.                 <div data-role=\"main\" class=\"ui-content\">\
120.                     \";\
121.                 if (digitalRead(rele1) == LOW)
122.                 {
123.                     web += \"\
124.                         <div class=\"boxed\" style=\"background-color:rgb(60, 179, 113); text-align:cen-
125.                         ter; vertical-align: middle; width:50px; height:50px;display: inline-block;border-ra-
126.                         dius: 40%;\">\
127.                         <h1> P1 </h1>\
128.                     </div>\
129.                     \";\
130.                 }
131.             </div>\
132.         </body>\
133.     </html>";

```

```

128. {
129.     web += "\
130.         <div class=\"boxed\" style=\"background-color:rgb(255, 0, 0); text-align:cen-
            ter; vertical-align: middle; width:50px; height:50px;display: inline-block;border-ra-
            dius: 40%;\">\
131. <h1> P1 </h1>\
132.</div>\
133.         ";
134. }
135. if (digitalRead(rele2) == LOW)
136. {
137.     web += "\
138.         <div class=\"boxed\" style=\"background-color:rgb(60, 179, 113); text-align:cen-
            ter; vertical-align: middle; width:50px; height:50px;display: inline-block;border-ra-
            dius: 40%;\">\
139. <h1> P2 </h1>\
140.</div>\
141.         ";
142. }
143. else
144. {
145.     web += "\
146.         <div class=\"boxed\" style=\"background-color:rgb(255, 0, 0); text-align:cen-
            ter; vertical-align: middle; width:50px; height:50px;display: inline-block;border-ra-
            dius: 40%;\">\
147. <h1> P2 </h1>\
148.</div>\
149.         ";
150. }
151. if (digitalRead(rele3) == LOW)
152. {
153.     web += "\
154.         <div class=\"boxed\" style=\"background-color:rgb(60, 179, 113); text-align:cen-
            ter; vertical-align: middle; width:50px; height:50px;display: inline-block;border-ra-
            dius: 40%;\">\
155. <h1> P3 </h1>\
156.</div>\
157.         ";
158. }
159. else
160. {
161.     web += "\
162.         <div class=\"boxed\" style=\"background-color:rgb(255, 0, 0); text-align:cen-
            ter; vertical-align: middle; width:50px; height:50px;display: inline-block;border-ra-
            dius: 40%;\">\
163. <h1> P3 </h1>\
164.</div>\
165.         ";
166. }
167.
168. if (digitalRead(rele4) == LOW) {
169.     web += "\
170.         <div class=\"boxed\" style=\"background-color:rgb(60, 179, 113); text-align:cen-
            ter; vertical-align: middle; width:50px; height:50px;display: inline-block;border-ra-
            dius: 40%;\">\
171. <h1> P4 </h1>\
172.</div>\
173.         ";
174. }
175. else
176. {
177.     web += "\
178.         <div class=\"boxed\" style=\"background-color:rgb(255, 0, 0); text-align:cen-
            ter; vertical-align: middle; width:50px; height:50px;display: inline-block;border-ra-
            dius: 40%;\">\
179. <h1> P4 </h1>\
180.</div>\

```

```

181.         ";
182.     }
183.     web += "\
184.     <br>\
185.     <a href=\"#overlayPanel\" class=\"ui-btn ui-icon-alert ui-btn-icon-left ui-shadow\">Pis-
186.         torasiat</a>\
187.         <div data-role=\"collapsible\">\
188.             <div data-role=\"panel\" id=\"overlayPanel\" data-display=\"overlay\">\
189.                 ";
190.                 web += "\
191.                 <div data-role=\"controlgroup\" data-type=\"horizontal\">\
192.                     <h1><p>PISTORASIA 1</p></h1>\
193.                 ";
194.
195.                 if ( digitalRead(rele1)==LOW ) {
196.                     web += "\
197.                         <a href=\"/?pistoke1=off\" class=\"ui-btn ui-icon-delete ui-btn-icon-left ui-
198.                             btn-inline ui-btn-b\">OFF</a>\
199.                     ";
200.                 }
201.                 else {
202.                     web += "\
203.                         <a href=\"/?pistoke1=on\" class=\"ui-btn ui-icon-check ui-btn-icon-left ui-
204.                             shadow\">ON</a>\
205.                     ";
206.                 }
207.                 web += "\
208.                 </div>\
209.                 <div data-role=\"controlgroup\" data-type=\"horizontal\">\
210.                     <h1><p>PISTORASIA 2</p></h1>\
211.                 ";
212.                 if ( digitalRead(rele2)==LOW ) {
213.                     web += "\
214.                         <a href=\"/?pistoke2=off\" class=\"ui-btn ui-icon-delete ui-btn-icon-left ui-
215.                             btn-inline ui-btn-b\">OFF</a>\
216.                     ";
217.                 }
218.                 else {
219.                     web += "\
220.                         <a href=\"/?pistoke2=on\" class=\"ui-btn ui-icon-check ui-btn-icon-left ui-
221.                             shadow\">ON</a>\
222.                     ";
223.                 }
224.                 web += "\
225.                 </div>\
226.                 <div data-role=\"controlgroup\" data-type=\"horizontal\">\
227.                     <h1><p>PISTORASIA 3</p></h1>\
228.                 ";
229.                 if ( digitalRead(rele3)==LOW ) {
230.                     web += "\
231.                         <a href=\"/?pistoke3=off\" class=\"ui-btn ui-icon-delete ui-btn-icon-left ui-btn-
232.                             inline ui-btn-b\">OFF</a>\
233.                     ";
234.                 }
235.                 else {
236.                     web += "\
237.                         <a href=\"/?pistoke3=on\" class=\"ui-btn ui-icon-check ui-btn-icon-left ui-
238.                             shadow\">ON</a>\
239.                     ";
240.                 }
241.                 web += "\
242.                 <div data-role=\"controlgroup\" data-type=\"horizontal\">\
243.                     <h1><p>PISTORASIA 4</p></h1>\

```

```

241.     ";
242.     if ( digitalRead(rele4)==LOW ) {
243.         web += "\
244.             <a href=\"/?pisto4=off\" class=\"ui-btn ui-icon-delete ui-btn-icon-left ui-btn-
inline ui-btn-b\">OFF</a>\
245.         ";
246.     }
247.     else {
248.         web += "\
249.             <a href=\"/?pisto4=on\" class=\"ui-btn ui-icon-check ui-btn-icon-left ui-
shadow\">ON</a>\
250.         ";
251.     }
252.     web += "\
253.         </div>\
254.     </div>\
255. </div>\
256. <div data-role=\"footer\">\
257.     <h1>© Amir Dirbaz - 1.0V - 2018 </h1>\
258. </div>\
259. </body>\
260. </html>";
261. server.send ( 200, "text/html", web );
262.
263. }
264. void ip(void) {
265.     lcd.setCursor(0,0);
266.     lcd.print("SSID: SMARTBOX");
267.     lcd.setCursor(0,1);
268.     lcd.print("PW: 123456789BOX");
269. }
270. void painik1()
271. {
272.     if (digitalRead(painike1)==LOW){
273.         if (digitalRead(rele1)==HIGH) {
274.             digitalWrite(rele1, LOW);
275.         }
276.         else
277.         {
278.             digitalWrite(rele1, HIGH);
279.         }
280.     }
281.     if (digitalRead(painike2)==LOW){
282.         if (digitalRead(rele2)==HIGH) {
283.             digitalWrite(rele2, LOW);
284.         }
285.         else
286.         {
287.             digitalWrite(rele2, HIGH);
288.         }
289.     }
290. }
291. if (digitalRead(painike3)==LOW){
292.     if (digitalRead(rele3)==HIGH) {
293.         digitalWrite(rele3, LOW);
294.     }
295.     else
296.     {
297.         digitalWrite(rele3, HIGH);
298.     }
299. }
300. if (digitalRead(painike4)==LOW){
301.     if (digitalRead(rele4)==HIGH) {
302.         digitalWrite(rele4, LOW);
303.     }
304.     else
305.     {

```

```
306.    digitalWrite(rele4, HIGH);  
307.  }  
308.  }  
309.  delay(200);  
310. }
```

